RUTDevKit User Manual

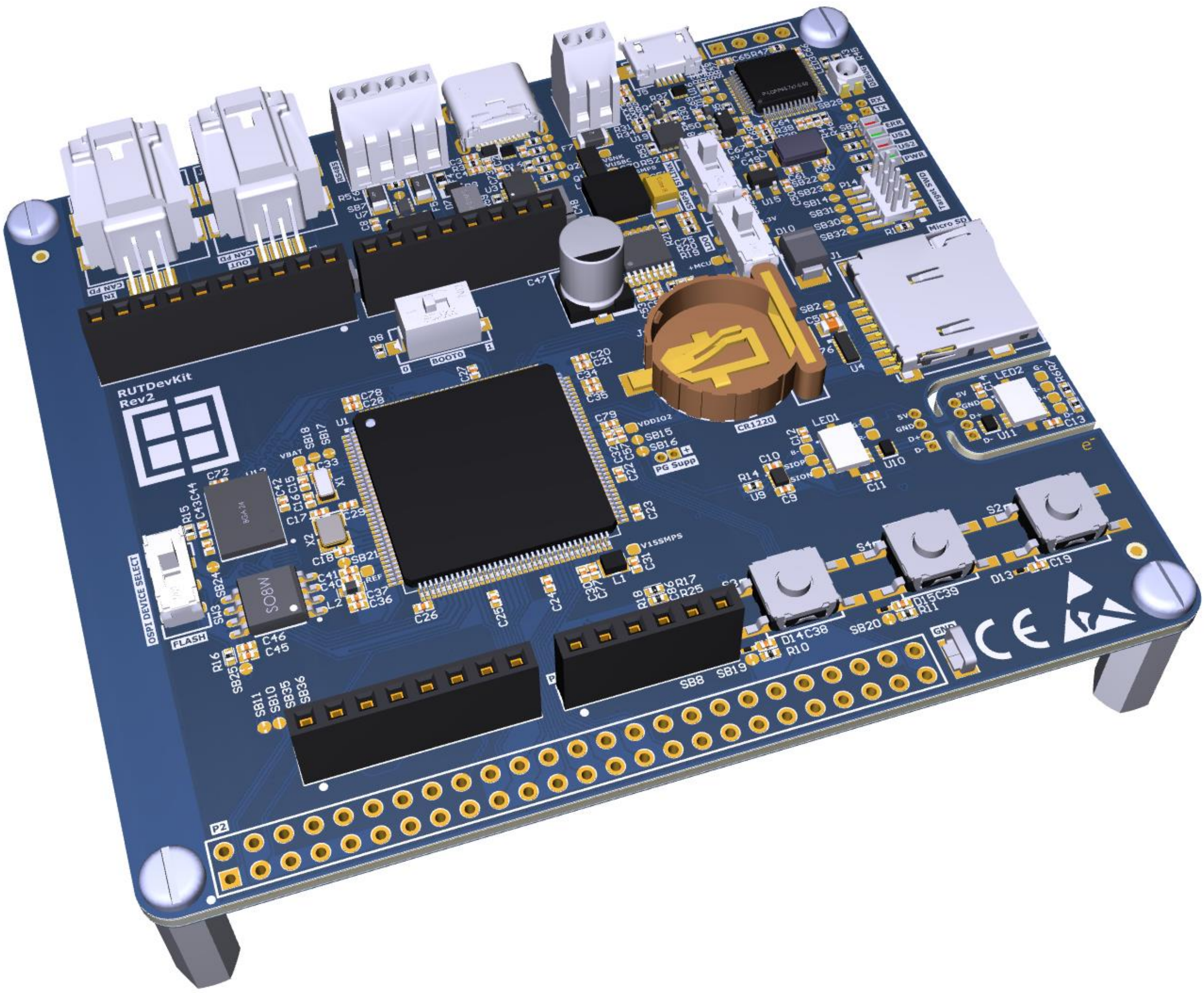

Table of Contents

Versions

Table 1

Version	Date	Rationale
0.1	November 14, 2019	First draft.
1.0	May 04, 2020	Rev. 2 Release.

Introduction

RUTDevKit is a development platform used by firmware and hardware designers to develop their products. RUTDevKit was designed by Rutronik to promote outstanding products selected only from their suppliers.

Features

- STMicroelectronics STM32L562ZET6Q Cortex®-M33 512KB Flash 256KB SRAM Ultra Microcontroller.
- External AP Memory APS6408L 64Mbit Octo-SPI PSRAM.
- External ESMT EN25QH128A 128Mbit Quad-SPI NOR Flash.
- USB Power Delivery Sink with TCPP01-M12 protection IC.
- STMicroelectronics SMPS L6986 38V Max. Input, 2A Max. Output.
- CAN FD with TLE9251VLE driver from Infineon.
- RS485 interface with ST3485EDBR driver from STMicroelectronics.
- Adam-Tech Micro SD Card Socket.
- On-Board ST-Link.
- ARM JTAG Header: 10-pin, 1.27mm pitch, 2 rows for Target MCU.
- Arduino compatible headers.
- STMicroelectronics TVS and ESD protectors.
- Amphenol FCI MINITEK µSPACE Connectors for CAN FD.
- Two user buttons and one for Reset from C&K.
- OSRAM LED Indicators
- Keystone CR1220 battery holder for low power mode tests.
- DIPSWITCH from Diptronics for MCU BOOT Select.
- CHILSIN Power Inductors.
- Quartz Oscillators from Epson and HKC.

Overview

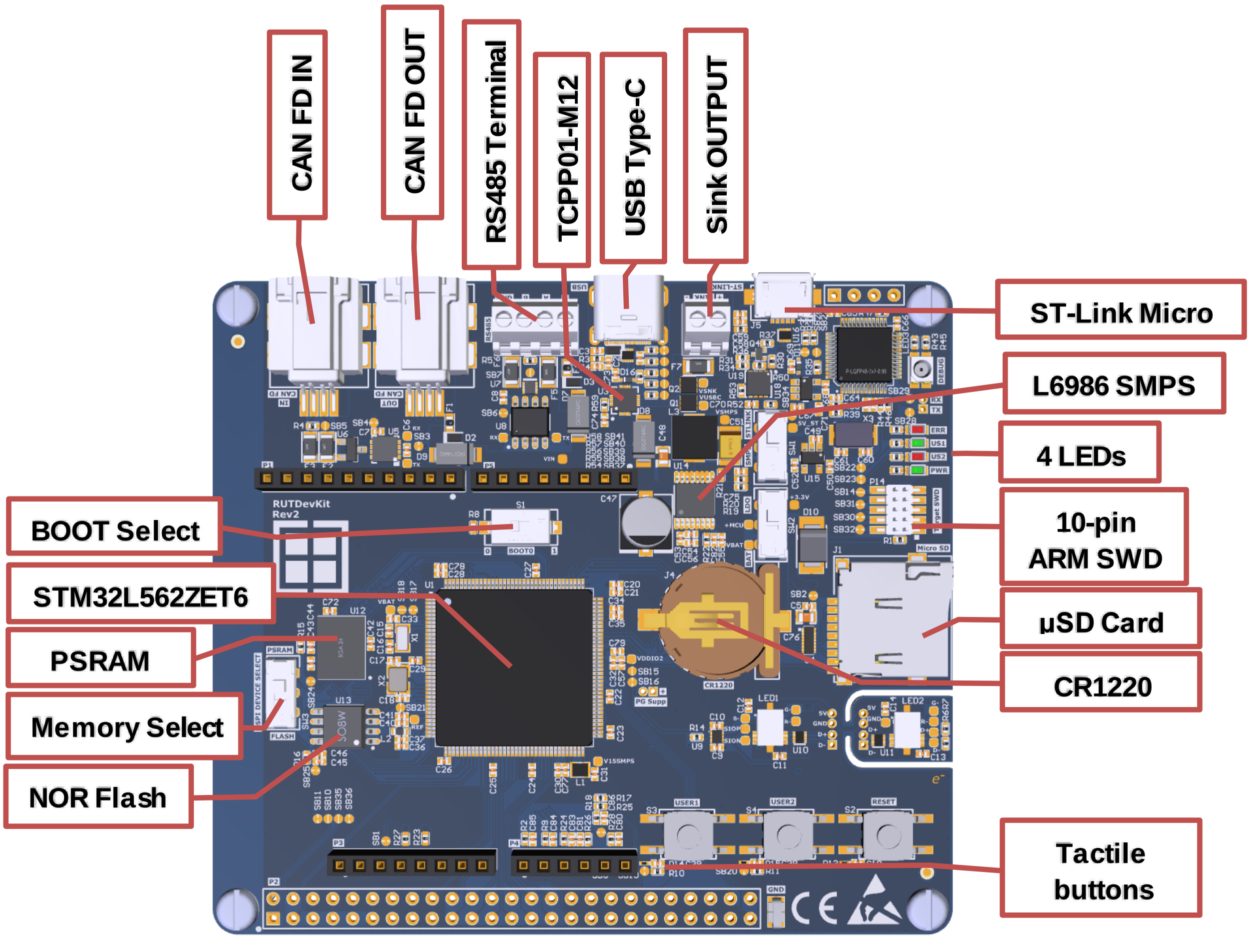

Fig. 1. RUTDevKit Evaluation board's layout.

BOOT Select

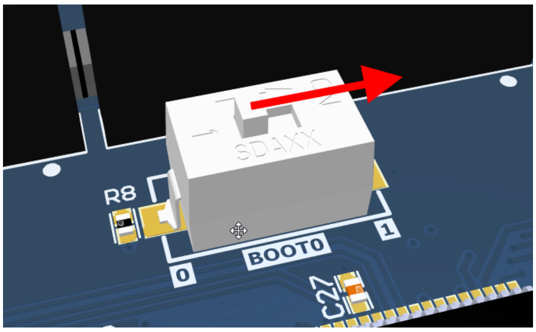

Fig. 2. Move the switch to the right to enable the system bootloader.

USB DFU or UART1 bootloader mode is available if switch S1 is in the "1" position. Please slide the switch S1 to the right "1" and press the S2 RESET button to activate the bootloader. The firmware detects automatically if a USB cable is present and enters into DFU mode. If a USB cable is not present, the UART1 bootloader will be activated and it is accessible via ST-Link's VCOM. For more information about system bootloader please refer to Application note AN2606.

Memory Select

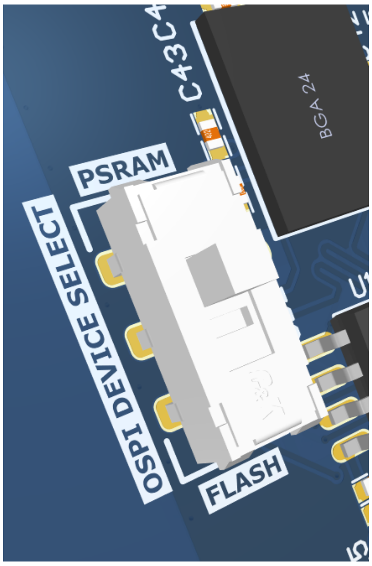

Fig. 3. Memory selector.

Since the STM32L5 MCUs have only one OSPI interface the RUTDevKit was designed to use only one OSPI device at a time: OSPI PSRAM or QSPI NOR FLASH. Use SW3 to select between these two devices. SW3 Switch selects CS signal available for MCUs OSPI interface. Unused device's CS pin remains pulled up to VCC, hence not selected state.

Power Source Select

Four power sources are available in RUTDevKit:

1. ST-Link USB port.
2. SMPS powered from CAN FD, RS485, and USB Type C interfaces.
3. CR1220 coin battery socket.
4. Arduino connectors – configured using R23 and R27 0R 0402 resistors.

Select the power source using SW1 STLINK or SMPS. With SW2 users can select the power source as BAT – coin battery or LDO 3.3V. LDO is powered from SW1 selected power source.

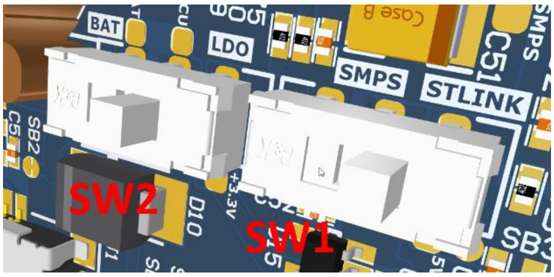

Fig. 4. Power source selectors.

Programming Using External Connector

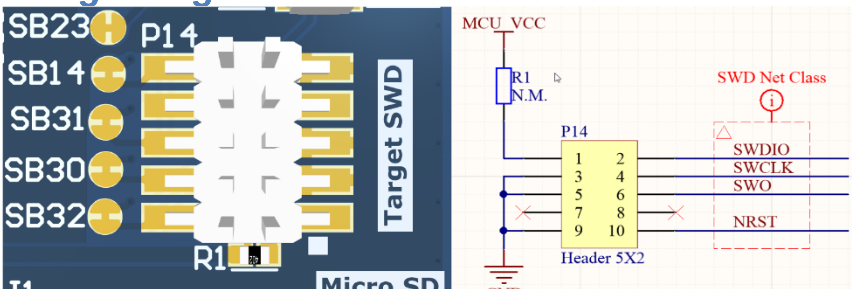

Fig. 5. 10-pin male 1.27mm pitch, SWD connector.

Users may use third party programming devices to connect STM32L562 target via P14 SWD connector. By default target, MCU is connected to ST-Link debugger. To use external SWD port the solder bridges have to be disconnected: SB30, SB31, SB32.

CAN FD Sockets

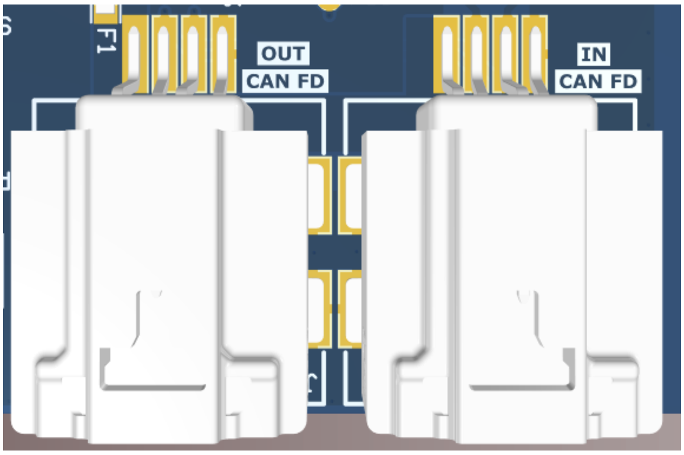

Fig. 6. CAN FD Connectors.

Two Amphenol ICC Minitek MicroSpace™ are used for CAN FD connection. Part No.: **10142344-104KLF** . One for CAN input cable, another for CAN output cable. To connect the wires certain receptacles are needed: 10142348-004LF and contacts 10141272-111LF have to be crimped and mounted into the receptacle.

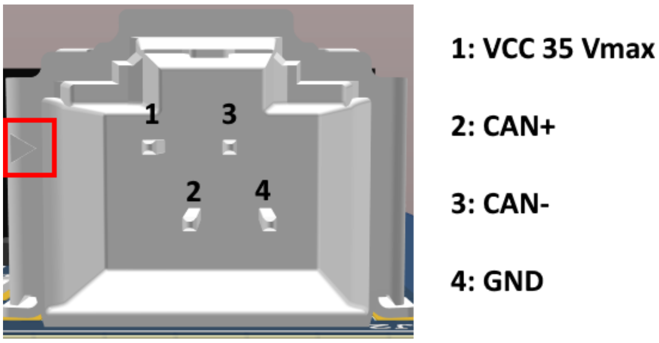

Fig. 7. CAN FD Pinout

Spare GPIOs

All unused GPIOs of STM32L562 MCU are available at socket P2. Some may need to be configured using solder bridges.

Fig. 8. P2 Socket for spare GPIOs

Table 2

Socket P2 Pinout

Pin No.	Name	Name	Pin No.
1	PA5	-	2
3	PB2	PB7	4
5	PB6	PB8	6
7	PB4	PB9	8
9	PD14	-	10
11	PC13	PC6	12
13	PD10	PD11	14
15	PG10	-	16
17	PG12	PF11	18
19	PF13	PF12	20
21	PF15	PF14	22
23	PG1	PG0	24
25	PE7	PE10	26
27	PE11	PE12	28
29	PE13	PE14	30
31	PE15	PD15	32
33	PF10	PF9	34
35	PF8	PF7	36

37	PF4	PF3	38
39	PF2	PF1	40
41	PF0	PE6	42
43	PE5	PE4	44
45	PE2	PE1	46
47	PE0	PG6	48
49	PG7	PG8	50

Solder Bridges

Table 3

Solder Bridge	Circuit	Default
SB1	MCU RESET Signal with Arduino RESET Input.	Closed
SB2	MCU VCC for micro SD card.	Closed
SB3	5V for CAN FD Driver.	Closed
SB4	MCU VCC for CAN FD Driver.	Closed
SB5	CAN FD 120 Ohm Termination.	Open
SB6	MCU VCC for RS485 Driver.	Closed
SB7	RS485 120 Ohm Termination.	Open
SB8	USBPD Voltage divider with ADC12_IN7.	Closed
SB10	PA5 GPIO with P2 Socket.	Open
SB11	USBPD Dead Battery signal with PA5 GPIO.	Closed
SB14	MCU RESET Signal with MCU Pin NRST.	Open
SB15	External voltage for PORTG.	Open
SB16	MCU VCC for PORTG.	Closed
SB17	MCU VCC with VBAT input.	Closed
SB18	Battery Socket with VBAT.	Open
SB19	USER1 Button with PB6 GPIO.	Closed
SB20	USER2 Button with PB7 GPIO.	Closed
SB21	MCU VCC with VREF+ input.	Closed
SB22	D5 LED with PB8 GPIO.	Closed
SB23	D6 LED with PB9 GPIO.	Closed
SB24	MCU VCC for external PSRAM.	Closed
SB25	MCU VCC for external NOR Flash.	Closed
SB26	External +5V power detection for ST-Link.	Closed
SB27	External +3.3V power detection for ST-Link.	Open
SB28	USART1 RX with ST-Link VCOM TX.	Closed
SB29	USART1 TX with ST-Link VCOM RX.	Closed
SB30	MCU SWD CLK with ST-Link Debugger.	Closed
SB31	MCU SWD SWO with ST-Link Debugger.	Closed
SB32	MCU SWD DIO with ST-Link Debugger.	Closed
SB34	ST-Link power source current limiting bypass.	Open
SB35	PC13 GPIO with P2 Socket.	Open
SB36	USBPD Fault Detection with PC13 GPIO.	Closed

SB37	USBPD Cut-off Voltage 6V.	Open
SB38	USBPD Cut-off Voltage 10V.	Open
SB39	USBPD Cut-off Voltage 13V.	Open
SB40	USBPD Cut-off Voltage 17V.	Open
SB41	USBPD Cut-off Voltage 22V.	Closed

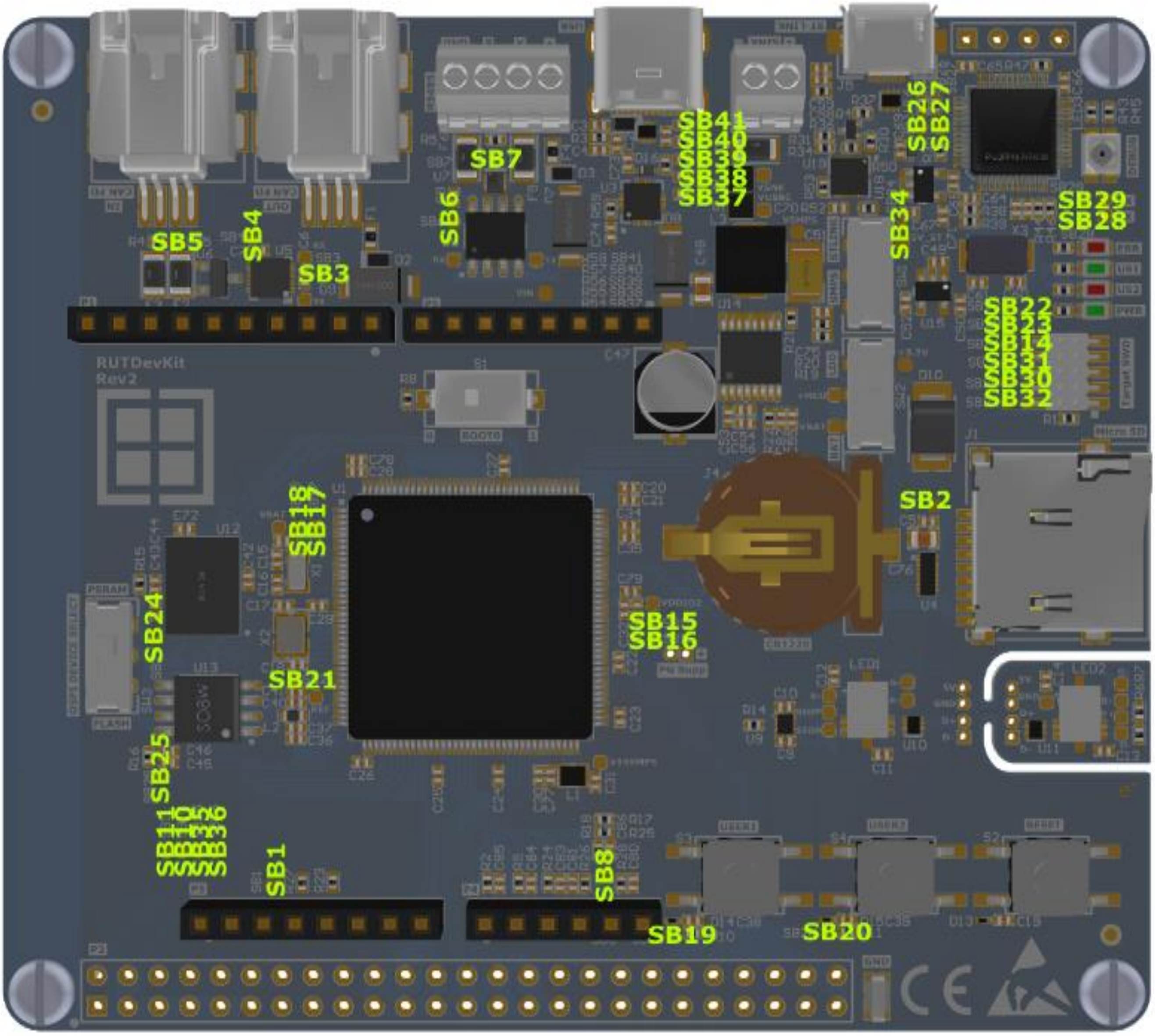

Fig. 9. Locations of the Solder Bridges.

Fuses

Resettable and non-resettable fuses are used for this project. In case the fuses are not fit for the user's final application the user has to change the fuses by unsoldering and soldering new ones that meet the requirements. Rev2 fuses list:

1. **F1, F4** “Super-Quick-Acting” 2A, 32V 0603 SMD. Part No.: 3412.0119.11.
2. **F2, F3, F5, F6** “Resettable PTC” 50mA 60V 1206 SMD. Part No.: PTS120660V005.
3. **F7** “High I^2t Chip” 5A, 32V 1206 SMD. Part No.: CC12H5A-TR.

Demo Firmware Examples

Default Application

The New board comes with already preprogrammed firmware. This firmware is used for hardware testing/diagnostics. The whole testing process is monitored using STM32CubeMonitor-UCPD software. Use the “TRACES” option to monitor outgoing messages from ST-Link VCOM port.

How to use:

1. Have STM32CubeMonitor-UCPD installed on your Laptop/PC.
2. Launch STM32CubeMonitor-UCPD software.
3. Connect RUTDevKit to ST-Link’s Micro USB port.
4. Press the “RESET” button on the board and TRACES→Select ST-Link VCOM in STM32CubeMonitor-UCPD software. Please do this step in 5 seconds.
5. The terminal window should show progress with all the tests beginning from PSRAM Test as shown below:

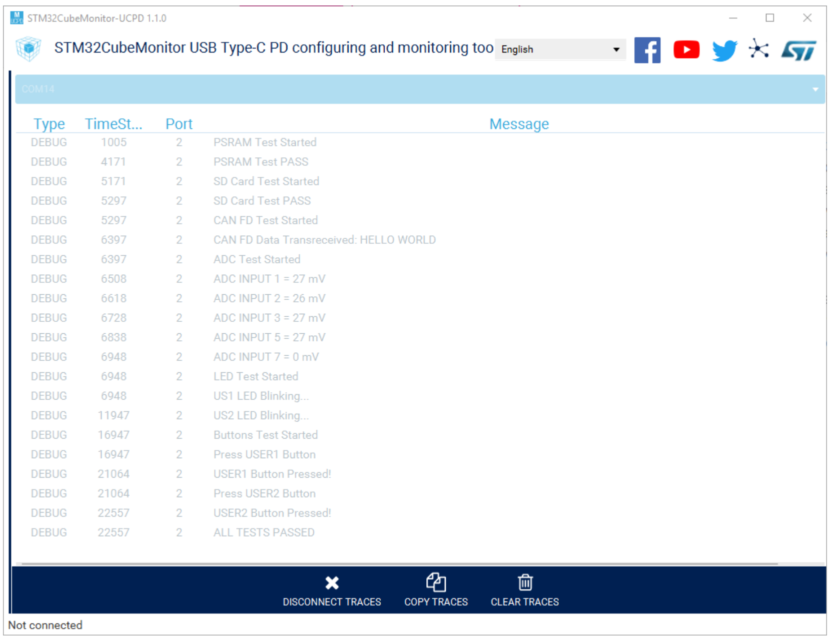

Fig. 10. RUTDevKit Hardware Tests Passed.

RS485 and LPUART ports on RUTDevkit will echo any character you enter using your preferred terminal software, PuTTY, RealTerm, etc.

The application will stop on any hardware failure except Micro SD Card Test if it is not inserted the test will be skipped. Format Card with FAT32 before testing.

Telit Modem & Telit Cloud Demo

This application introduces new users to Telit Cloud solution and Telit's NB-IoT/LTE-M modem ME310G1-W1.

The X-NUCLEO-IKS01A3 shield is used additionally to collect the data from sensors and upload it to the Telit IoT Portal periodically. Also, the ME310 modem shield with NB-IoT enabled SIM card is needed to run this application.

Access to Telit IoT portal with administrator rights is provided by the Telit https://www.telit.com/m2m-iot-products/iot-platforms/telit-iot-portal/ .

Telit IoT Portal view only account:

User name: rutronik_lt@rutronik.com

Password: Rutron1k_user

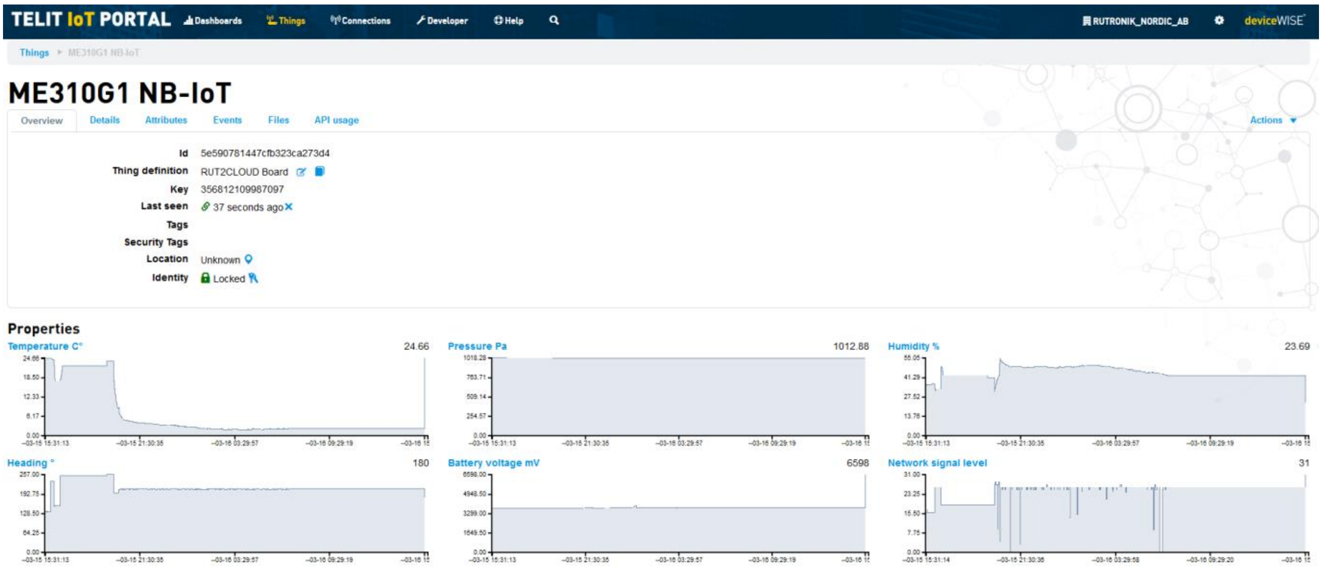

Fig. 11. Telit IoT Portal Things review.

AP Memory APS6408L-3OB test

The purpose of this test firmware is to demonstrate the performance of the PSRAM memory APS6408L-3OB-BA using the STM32L562 Octo-SPI peripheral. PSRAM is used in Direct Mapped Memory mode the whole time in this firmware example.

The program configures the Octo-SPI interface and PSRAM for maximum speed and then does the write/read operations to the PSRAM indefinitely. If any error is detected, it is indicated by the User LED US2.

ESMT NOR Flash test

The purpose of this test firmware is to demonstrate the performance of the NOR FLASH memory EN25QH128A-104HIP2T using the STM32L562 Octo-SPI peripheral in Quad-SPI mode. The memory is configured to work in Mapped Memory mode for reading operations only. For write operations, the memory is accessed in Indirect Quad-write mode.

Every time the MCU is powered up, the whole FLASH memory is erased and written with test data and then is read indefinitely looking for errors. If any error is detected, it is indicated by the User LED US2.

SD Card test

This firmware example is written to test the SD/MMC card interface.

The firmware configures the interface, mounts the media, and attempts to create a new test file „TEST.TXT“ filled with chars „A“. If the operation is successful then it reads the file checking for errors indefinitely. If any error is detected the Error_Handler() is called where two user LEDS US1 and US2 are blinking.

RS485 with MODBUS protocol

The purpose of this firmware example is to demonstrate the performance of the RS485 peripheral.

Since the MODBUS protocol is very mature and popular it was chosen to be implemented in this demo.

To use the RS485 MODBUS firmware example the RS485 USB adapter and any MODBUS PC software are needed. The free software was used for testing: „Radzio Modbus Master Simulator“.

The MODBUS is used in RTU mode. Eight Input Registers are prepared to be read with values: 11, 22, 33 .. 88. The addresses are assigned from 1 to 8.

How to test RS485 example:

1. Flash the example image to RUTDevKit using ST-Link or USB bootloader.
2. Download and install RS485 monitoring PC software (Radzio Modbus Master Simulator freeware).
3. Connect RS485 adapter to the PC and RS485 wires to RUTDevKit RS485 terminal. Double-check if wires „A“ and „B“ are connected properly.
4. Open PC application and go to Connection → Settings and configure the Modbus RTU settings as it is shown:

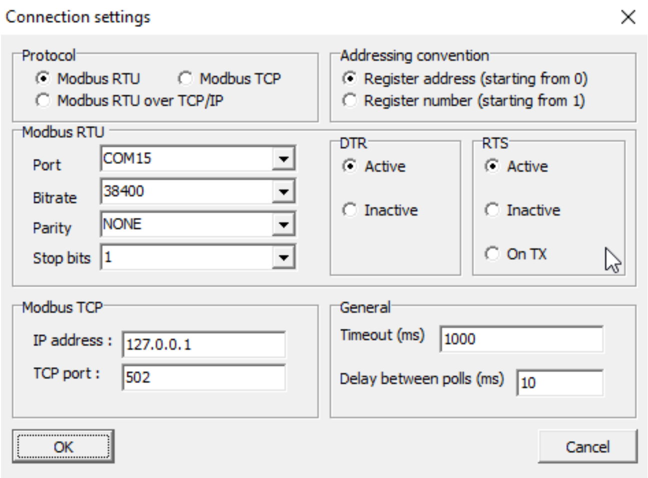

Fig. 12

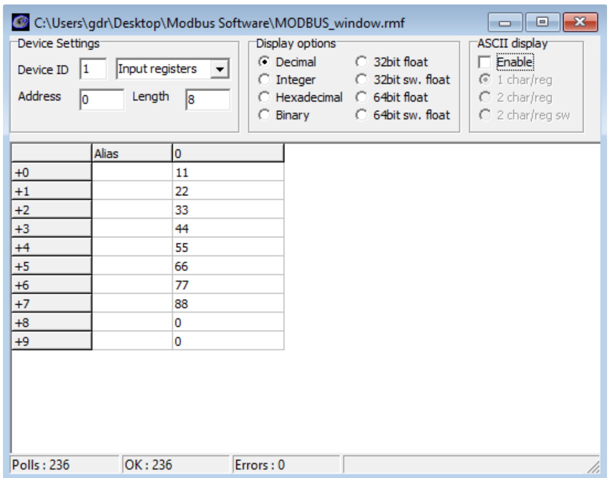

Fig. 13. Modbus Input Registers.

5. Press File → New and set up the window as shown below to be able to read the Input Registers. After you Press Connection → Connect the register values appears.

Arduino ADC channels using DMA0

This simple firmware example is used to read all ADC channels available in RUTDevKit from Arduino Analog inputs P4. The DMA Continues Conversion mode is used to read all the ADC channels. The ADC peripheral together with DMA is working without MCU intervention. Every time as the conversion is over the data is overwritten into the buffer adcBuffer[5];

Arduino I2C scanner

This simple example is used to test the I2C peripheral by scanning all the I2C's 7-bit addresses one by one and printing to ST-Link's VCOM if we have a response from any device online.

Dual Bank Flash on-the-fly programming

The purpose of this firmware example is to demonstrate Dual Flash bank capability in STM32L5 MCUs.

The demo is based on the STMicroelectronics Dual Bank demo for STM32L4+ MCUs (Application note AN4767).

The basic idea of this demo is to demonstrate how to program Bank B while running the tasks on Bank A and vice versa.

The most important difference for Dual Flash bank functionality between STM32L4 and STM32L5 is that by selecting the option bit in STM32L5: SWAP_BANK the bank addresses are swapped and MCU boots from swapped bank automatically at startup, hence there is no such a BFB2 bit (boot from Bank2) anymore in STM32L5 and no different addresses are needed to implement in firmware. From the application point of view, it always runs from Bank with the first address: 0x80000000.

How to use the example:

1. Install the Extra PuTTY on your PC.
2. Flash the Dual Bank example image to the RUTDevKit.
3. Open ST-Link VCOM with putty and press the RESET button on the RUTDevKit:

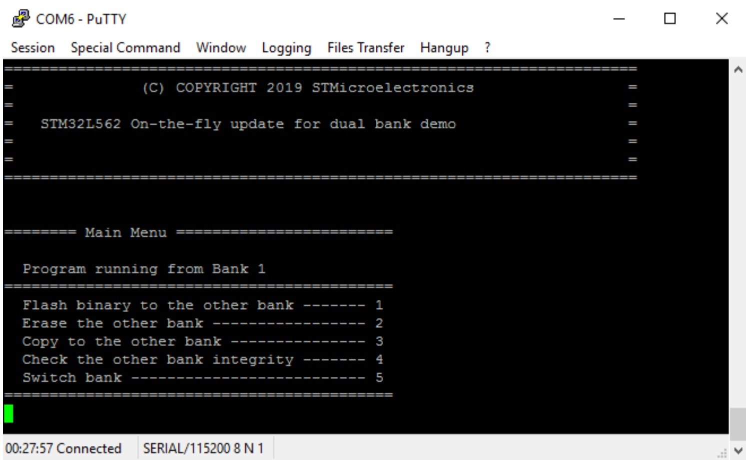

Fig. 14. CLI Menu of Dual Bank Example.

4. Enter option „1“ and select a binary file to upload Files Transfer → Ymodem → Send. Use the same Dual Bank example, but only in .bin format:

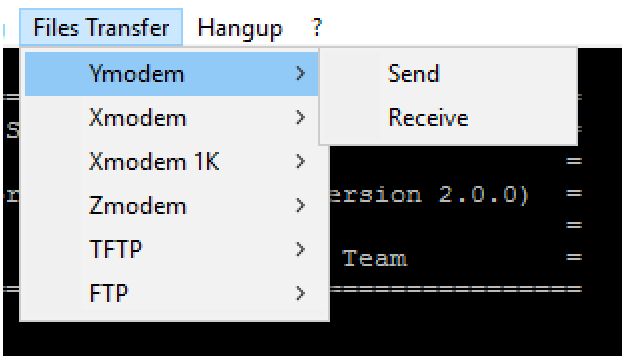

Fig. 15. YMODEM file transfer to RUTDevKit.

5. File transfer progress bar appears:

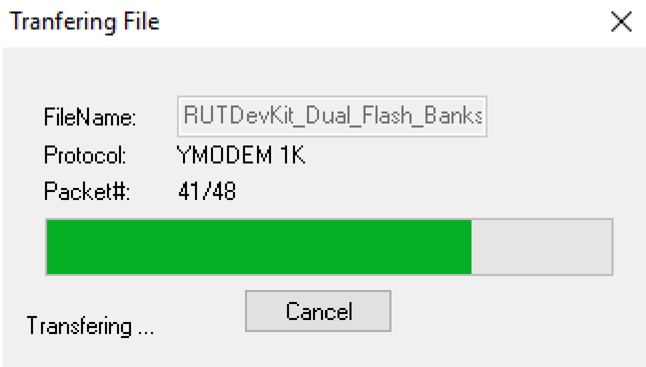

Fig. 16. Transfer to flash bank in progress.

6. After the file transfer is completed enter option „5“ to swap flash bank. The firmware runs from another flash bank now.

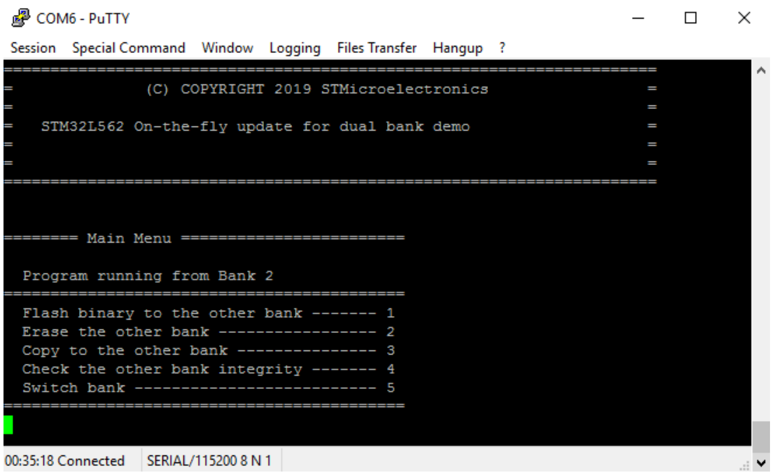

Fig. 17. Firmware is running from Bank 2.

CAN FD Test

A few firmware examples are prepared for RUTDevKit just to test CAN and CAN FD interface compatibility:

- CAN Standart Loopback Interrupt
- CAN Standart Loopback Polling
- CAN Standart Test
- CANFD Standart Loopback Interrupt
- CANFD Standart Loopback Polling
- CANFD Standart Test

The RUTDevKit's CAN and CAN FD compatibility were tested using STM32 Nucleo board with CAN/CANFD as well as in the loopback mode with RUTDevKit itself.

Tamper Protection Demo

This example is intended to demonstrate how to implement smart tamper protection for the products where opened housing could compromise the data or safety of the device.

Three tamper protection demo versions are available:

- Passive mode: edge detection
- Passive mode: level detection
- Active mode: comparator mode

The passive modes with edge and level detection simply detect the change of the voltage on the desired pin and that is treated as a security break.

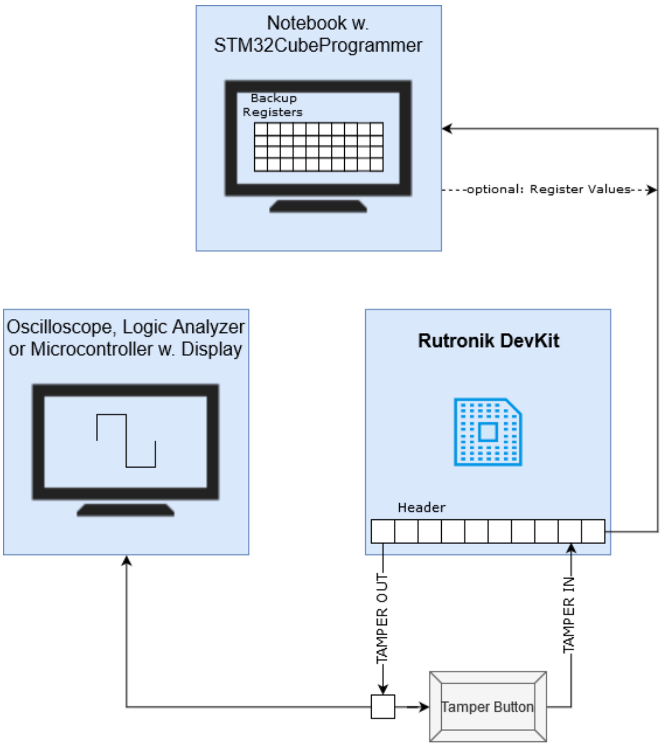

Fig. 18. Tamper protection demo setup.

The most secure is the active mode, where a (pseudo-)random generated square-wave signal is sent and compared. If the signal is cut off or distorted the content of the backup-registers will be deleted. The whole thing is part of the RTC peripheral so it may run powered from a coin cell battery for years.

If the user pushes the button the connection between the tamper output pin and tamper input pin will be blocked, an interrupt is fired and the registers cleared.

USB Power Delivery Test

The purpose of this firmware example is to demonstrate USB Power Delivery feature on the RUTDevKit board.

Test firmware initializes the USB PD hardware and configures the stack for communication with a power source. As soon the dialog with the power source is established the firmware selects the PDO – Power Data Object with highest voltage possible and applies it to the Sink Output. Hence the user may connect the load and evaluate the RUTDevKit capability of delivering the power to the load.

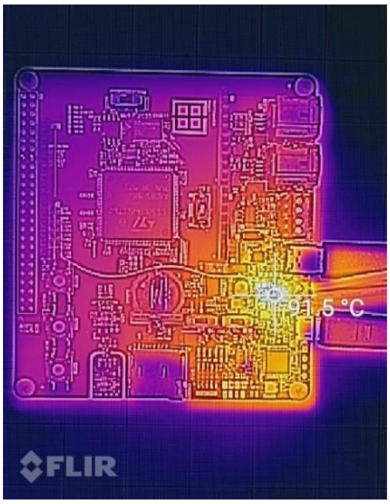

Fig. 19. RUTDevKit USB PD 20V 5A Sink output.

How to use the example:

1. Download and install the STM32CubeMonitor-UCPD software.
2. Flash RUTDevKit with USB PD example.

3. Open the STM32CubeMonitor, press icon „Trace“, then select the COM Port and press the „RESET“ button on the board, the message appears: USBPD_CAD_STATE_DETACHED

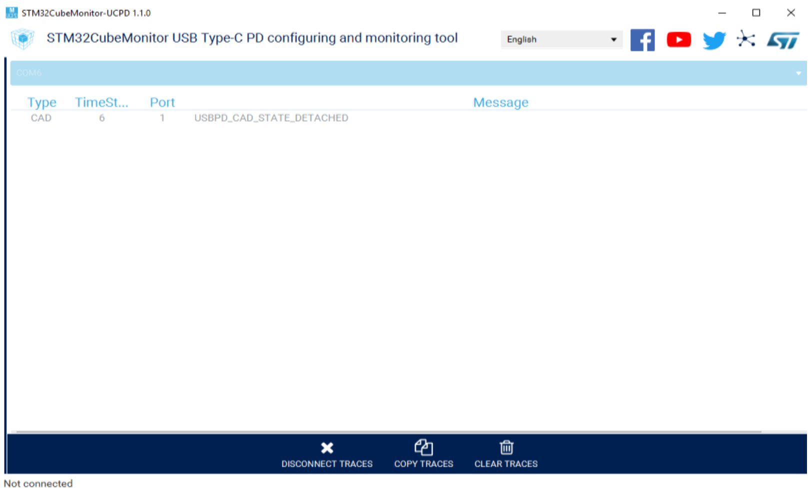

Fig. 20. STM32CubeMonitor Trace terminal ready.

4. Now connect the USB Type-C connector with the power source. Shortly the dialog appears and the most highest voltage is selected, the explicit power contract is established automatically:

Table 4

0	CAD	6	1	USBPD_CAD_STATE_DETACHED
1	CAD	536020	1	USBPD_CAD_STATE_ATTACHED_WAIT
2	CAD	536141	1	USBPD_CAD_STATE_ATTACHED0
3	EVENT	536141	1	EVENT_ATTACHED
4	PE	536141	1	PE_SNK_STARTUP
5	PE	536141	1	PE_SNK_WAIT_FOR_CAPABILITIES
6	IN	536161	1	SRC_CAPABILITIES DATA:F4910108F4D10200F4C10300F4B10400F4A10500F4410600 / 5V - 5A / 9V - 5A / 12V - 5A / 15V - 5A / 18V - 5A / 20V - 5A SOP PD2 H:0x6161
7	OUT	536162	1	GOODCRC SOP H:0x0041
8	PE	536162	1	PE_SNK_EVALUATE_CAPABILITY
9	PE	536164	1	PE_SNK_SEND_REQUEST
10	OUT	536164	1	REQUEST DATA: F4D10761 / ObjectPosition:6 / GiveBack:0 / CapabilityMismatch:0 / USBCommunicationCapable:0 / NoUSBSuspend:1 / UnchunkedExtendedMessagesSupported:0 SOP PD2 H:0x1042
11	IN	536165	1	GOODCRC SOP H:0x0161
12	PE	536165	1	PE_SNK_SELECT_CAPABILITY
13	IN	536166	1	ACCEPT SOP PD2 H:0x0363
14	OUT	536166	1	GOODCRC SOP H:0x0241
15	NOTIF	536166	1	POWER_STATE_CHANGE
16	NOTIF	536166	1	REQUEST_ACCEPTED

17	PE	536166	1	PE_SNK_TRANSITION_SNK		
18	IN	536225	1	PS_RDY SOP	PD2	H:0x0566
19	OUT	536225	1	GOODCRC	SOP	H:0x0441
20	NOTIF	536226	1	POWER_STATE_CHANGE		
21	NOTIF	536226	1	POWER_EXPLICIT_CONTRACT		
22	PE	536226	1	PE_STATE_READY		
23	NOTIF	536226	1	STATE_SNK_READY		
24	PE	536226	1	PE_STATE_READY_WAIT		

5. After the power negotiation dialog is over, the power is turned on at the SINK terminal of the board.

TrustZone for RUTDevKit Demo

The purpose of this firmware example is to introduce the user with TrustZone feature of Cortex-M33.

The TrustZone firmware template for STM32L562E-DK was used to build a demo for RUTDevKit. The firmware is split into parts: the secure and non-secure. The secure application always starts first, so it isolates and prepares the resources for non-secure application and then the non-secure application is started. If the user button USER1 is pressed while the non-secure application is running, the non-secure application tries to access protected RAM location and ends up in SecureFault_Handler.

How to get TrustZone example running on the RUTDevKit:

1. Before start working with TrustZone feature the option bit TZEN must be enabled. Also SECWM1_PSTRT/SECWM1_PEND and SECWM2_PSTRT/SECWM2_PEND should be set according to the application, use STM32CubeProgrammer to enable these options: TZEN=1, DBANK=1, SECWM1_PSTRT=0x0 SECWM1_PEND=0x7F, SECWM2_PSTRT=0x1 SECWM2_PEND=0x0.

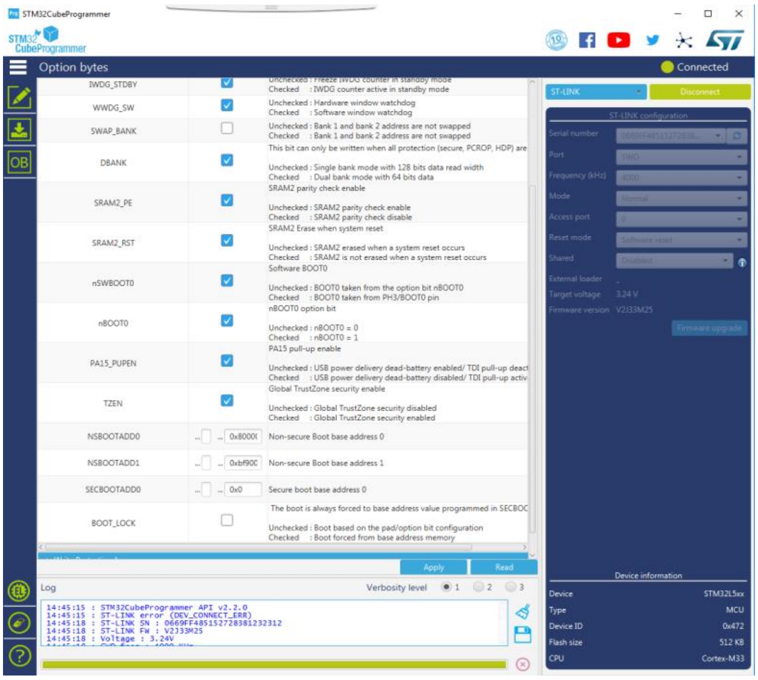

Fig. 21. TZEN enable using STM32CubeProgrammer.

2. Load the TrustZone example workspace with IAR Embedded Workbench IDE: File → Open Workspace…
3. Activate the secure project by selecting the “Project_s” tab and press “Download and Debug” to flash the project to the RUTDevKit.
4. Activate the non-secure project by selecting the “Project_ns” tab and press “Download and Debug” to flash the project to the RUTDevKit.

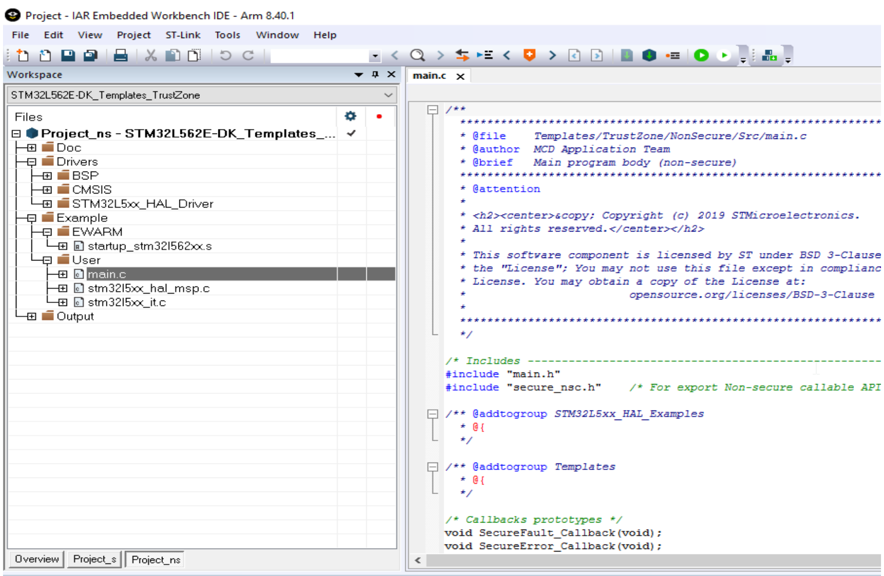

Fig. 22. TrustZone example loaded with IAR Workbench IDE.

5. While debugging a non-secure project, press the USER1 button, it will try to access restricted memory and the SecureFault_Handler() will be called.

Notice: To work with the RUTDevKit without TrustZone feature the TZEN must be disabled using this procedure:

1. Have a secure and non-secure application running properly.
2. Use STM32CubeProgrammer and CLI in windows to activate RDPLevel „1“:
   STM32_Programmer_CLI.exe -c port=swd mode=HOTPLUG -ob RDP=0xBB
3. Make sure the application is still running with RDP level „1“.
4. Do the regression deactivating at the same time TZ and RDP:
   `STM32_Programmer_CLI.exe -c port=swd mode=HOTPLUG -ob TZEN=0 RDP=0xAA`
5. You should have a full chip erase and TZEN=0 / RDP0 now.

Electromagnetic Compatibility

RUTDevKit was tested for electromagnetic disturbances and electromagnetic immunity. Also, RS485 and CAN FD interfaces were checked for immunity to conducted disturbances induced by radio-frequency fields and meets the requirements as in normative documents listed below:

Electromagnetic disturbances:
Radiated disturbance to 1 GHz
EN 55032:2015
EN 55032:2015/AC:2016-07

Electromagnetic immunity:
Electrostatic discharge immunity test
EN 55024:2010
EN 55024:2010/A1:2015
EN 61000-4-2:2009

Radiated RF electromagnetic field immunity test
EN 55024:2010
EN 55024:2010/A1:2015
EN 61000-4-3:2006
EN 61000-4-3:2006/A1:2008
EN 61000-4-3:2006/A2:2010

Immunity to conducted radio frequency electromagnetic disturbances (signal input/output ports)
EN 55024:2010
EN 55024:2010/A1:2015
EN 61000-4-6:2014

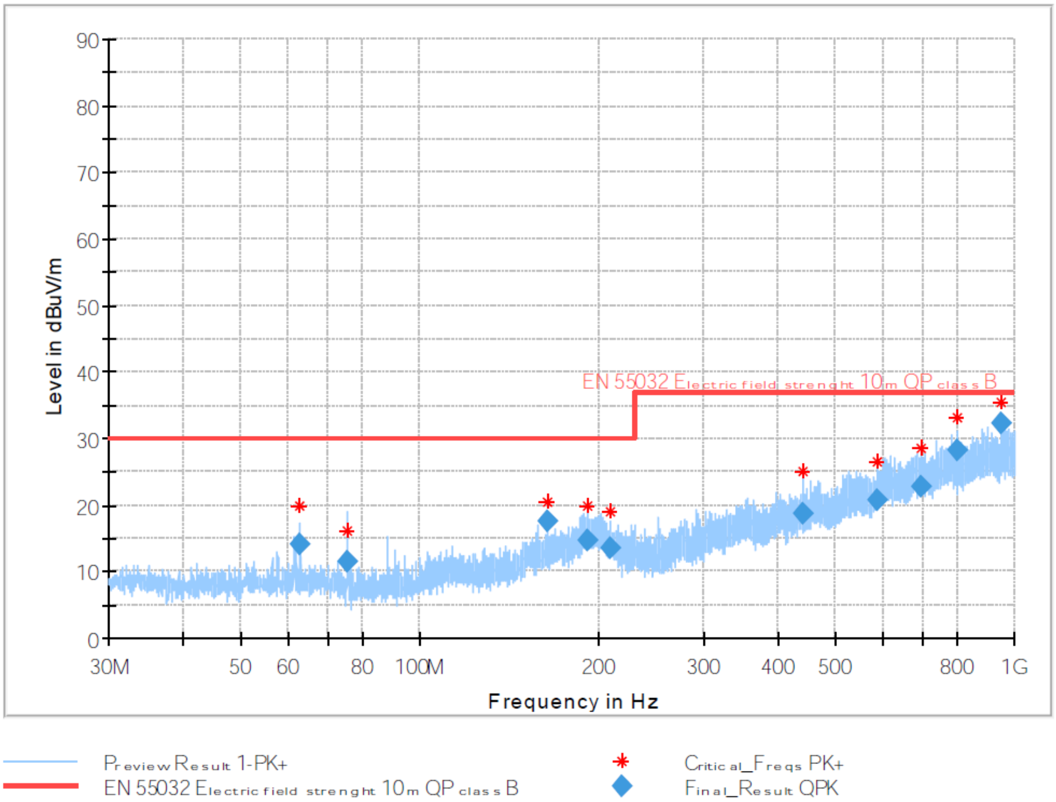

Fig. 23. Radiated disturbances while running AP Memory Test.

Legal Disclaimer

The evaluation board is for testing purposes only and, because it has limited functions and limited resilience, is not suitable for permanent use under real conditions. If the evaluation board is nevertheless used under real conditions, this is done at one's responsibility; any liability of Rutronik is insofar excluded.

Mechanical Layout